

Nanodegree Engenheiro de Machine Learning

Proposta de projeto final - Pokémon batalha - Usando Redes Neurais Temporais (Supervisionado)

Paulo Cotta
09 de Agosto de 2018

Proposta

A proposta do trabalho é executar processos onde os pokémons possam batalhar entre rodadas, sendo que cada treinador possa escolher apenas seis pokémons.

A ideia é criar cruzamentos entre os treinadores pokémons e validar quem teve a melhor performance entre todos os treinadores.

O DataSet escolhido foi retirado da plataforma Kaggle, conforme link a seguir: <https://www.kaggle.com/terminus7/pokemon-challenge/kernels>.

Segue uma introdução sobre o local que será utilizado:

Kanto (em japonês: カントー地方, Kantō-chihō) é um país ou região fictícia, da série Pokémon. Sua geografia é baseada na região de Kanto, uma região da ilha de Honshu, no Japão, de onde vem seu nome. A semelhança entre as formações de baía vistas no mapa do jogo e as formações reais de Sagami Bay, Suruga Bay e a Baía de Tokyo é particularmente impressionante. Kanto localiza-se a leste de Johto; presumivelmente, eles formam um pequeno continente.

Fonte: Wikipédia: [https://pt.wikipedia.org/wiki/Kanto_\(Pok%C3%A9mon\)](https://pt.wikipedia.org/wiki/Kanto_(Pok%C3%A9mon))

Os passos necessários para a execução do trabalho são:

Primeiro: preparação da base para que possa ter as especificações abaixo:

- Dados dos treinadores;
- Tipos de pokémons que cada treinador mais gosta;
- Qual região o treinador deu como seu ponto de partida;
- Sexo do treinador;
- Idade do treinador;
- Porcentagem de captura de pokémon.

Segundo: preparação dos treinadores de cada ginásio, seguem especificações abaixo:

- Idade do líder de ginásio;
- Sexo do líder de ginásio;
- Quais são os seus seis pokémons;
- Líder de qual cidade e ginásio;
- Ordem de batalhas do ginásio;
- Qual é o seu tipo de insígnia.

Criação do DataSet de treinadores e seus pokémons, conforme descrição abaixo:

- A partir do tipo de pokémon que o treinador mais gosta;

- A quilometragem da cidade de início para o primeiro ginásio e assim subsequente;
- Em cada cidade o percentual de ocorrência do pokémon;
- Criação de chance de captura;
- Caso o treinador já tenha atingido o limite de seis, o mesmo poderá evoluir o pokémon após cada batalha e, caso não o tenha, aumentar o percentual de poder.

Criação do DataSet de batalhas entre os treinadores e líderes de ginásio, deve seguir as seguintes situações:

- Batalhando a quantidade de seis pokémons do treinador contra os seis do líder;
- De uma cidade para outra o treinador pode alterar um pokémon;
- Quantas rodadas um treinador gastou para ganhar do líder;
- Melhorias de atributos pokémons após ganhar a insígnia;
- O clima do dia pode interferir na batalha.

O processo de ganhar uma insígnia faz com que os pokémons do treinador obtenham bônus, dependendo do tipo do pokémon o mesmo pode ganhar dobrado ou pode ganhar apenas o valor especificado.

- Tipos compatíveis com a insígnia ganham X2;
- Tipos não compatíveis com a insígnia ganham X1.

Criação do aprendizado de possível vitória do treinador na fase Liga Pokémon, onde um modelo neural irá avaliar os seguintes pontos, a cada cidade que o treinador passou, e gerar um nível de acurácia e uma taxa de perda. Os itens abaixo serão considerados:

- A cada cidade, a quantidade de vezes que executou uma batalha;
- Se repetiu mais de uma vez, é aplicada uma penalidade ao treinador;
- A cada troca de pokémon, ao treinador andar de uma cidade à outra.

No final, haverá as batalhas entre os treinadores, para verificar o ganhador da Liga Pokémon, lembrando que já foi previsto o vencedor, mas neste caso será utilizado um processo estatístico para validar e verificar o ganhador. Neste item será gerada uma saída através de um DataSet com estas informações.

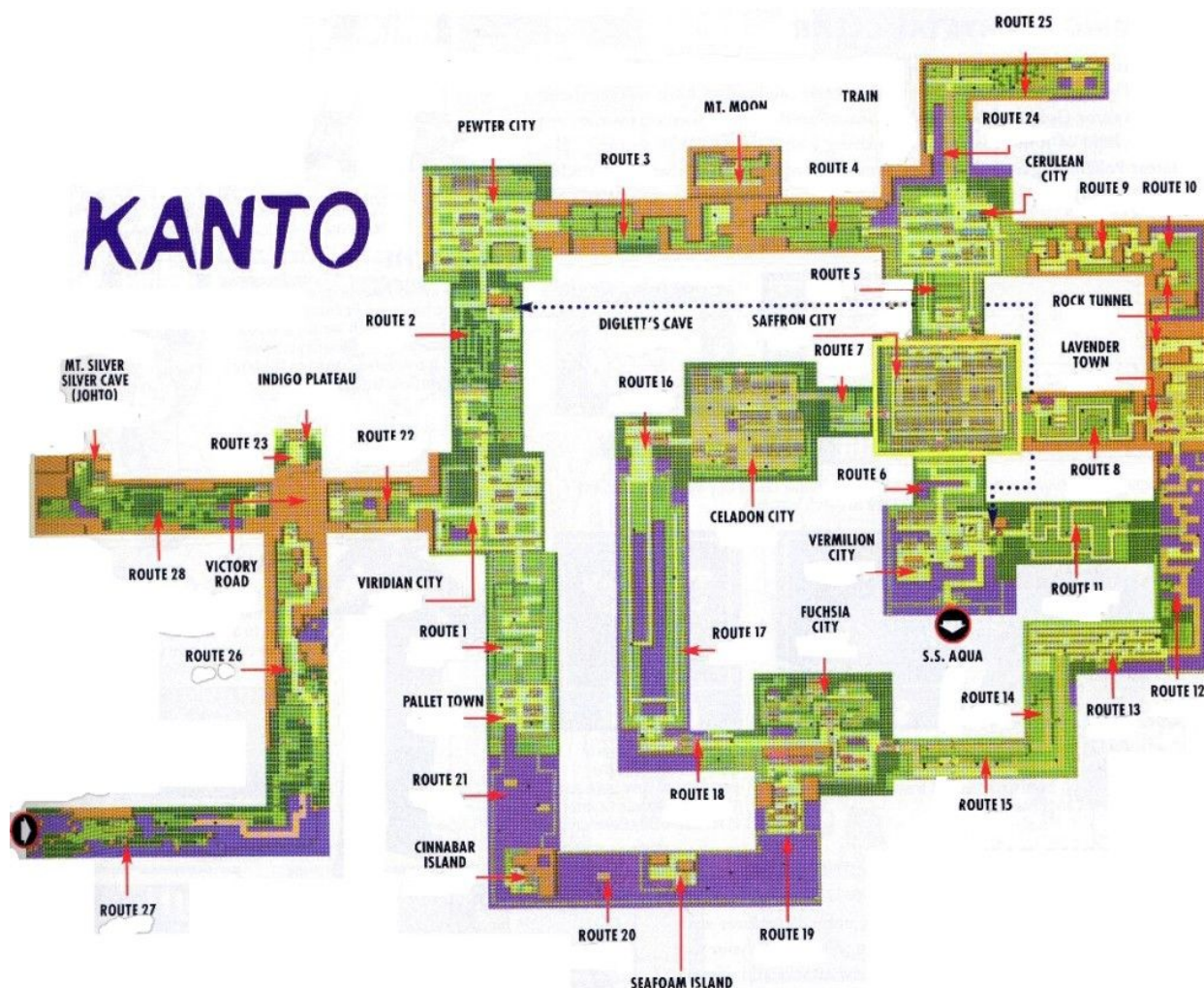
Será utilizado um modelo que possa verificar se o treinador irá conquistar o bônus, que é capturar um pokémon lendário e qual pokémon ele conseguiria capturar em caso positivo.

Listas das insígnias (São todos levando em consideração a região fictícia Kanto):

- Insígnia da Rocha;
- Insígnia da Cascata;
- Insígnia do Trovão;
- Insígnia do Arco-Íris;
- Insígnia da Alma;
- Insígnia da Lama;
- Insígnia do Vulcão;

- Insígnia da Terra.

Abaixo segue o mapa que será percorrido nos DataSet dos treinadores e as conquistas de insígnias.



Histórico do assunto

O histórico levantado é baseado em informações do Pokémon Go e dos jogos da franquia. A motivação pessoal é que pode ser realizada uma forma de jogo, baseada em Redes Neurais, que possa antecipar o processo de vitória de um treinador e, a cada frequência de movimentação do treinador, o seu fluxo temporal pode ser alterado para favorecer sua vitória final ou não.

A ideia é poder criar um mini jogo, onde apenas a máquina possa interagir de forma bem sentimental, de acordo com o clima, tipo pokémon, seus poderes e quantidade de vezes que perdeu para um oponente. Todas essas variáveis podem influenciar diretamente nas saídas.

Descrição do problema

O problema a resolver será o quanto uma rede temporal pode chegar a uma saída de acurácia e predição de ganhador, com interferências em variáveis mutáveis para cada acontecimento dos dados, exemplo: o clima que será randômico, os seis pokémons dos líderes contra os seis pokémons dos treinadores, onde o dos treinadores será capturado a partir de cada região de saída.

A ideia é criar uma rede em que seja possível efetuar uma predição assertiva e que consiga avaliar e identificar o ganhador final.

Conjuntos de dados e entradas

O conjunto de entrada, conforme descrito acima, foi retirado da plataforma Kaggle, o mesmo possui os 150 primeiros pokémons, com os quais serão trabalhados.

B	C	D	E	F	G	H	I	J	K	L
Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	FALSE

O outro DataSet será o de treinadores, que possuirá os seguintes campos:

- Dados dos treinadores;
- Tipos de pokémons que cada treinador mais gosta;
- Qual região o treinador deu como seu ponto de partida;
- Sexo do treinador;
- Idade do treinador.

Os dados são relevantes para o processo de limpeza, treino supervisionado na base de pokémons e não supervisionado na captura de pokémons para os treinadores. Dessa forma, coloca-se em prática o que foi apresentado no curso realizado na Udacity.

Descrição da solução

A solução permitirá criar um processo de rede neural temporal, onde possa prever algumas situações dos treinadores a partir de saídas randômicas.

O processo inicia com redes temporais regulares, onde o DataSet está organizado e definido, esperando o processo evolutivo do seu padrão.

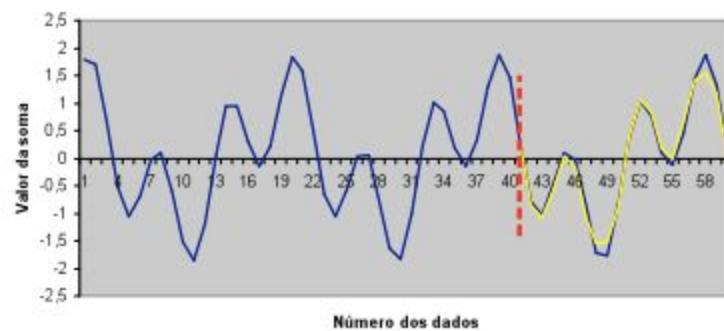
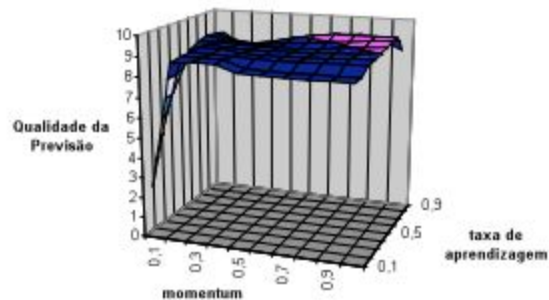
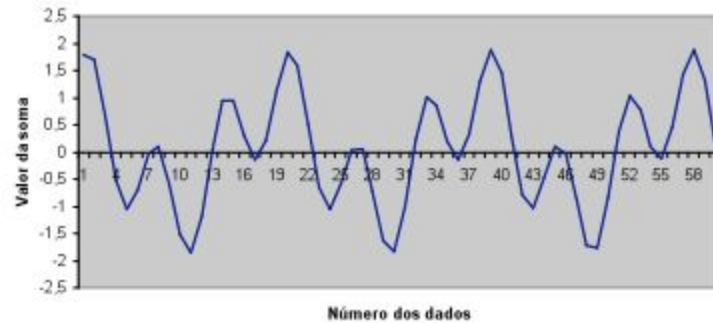
Após a captura de pokémons, a rede temporal será irregular, onde o processo inicia como regular, mas algumas variáveis alteram o seu processo e suas porcentagens.

Modelo de referência

Os modelos de dados que serviram de referência são os modelos de predição usados por sistemas que tentam prever a bolsa de valores, ou finanças para seus usuários finais.

Como exemplo, pode-se citar o Banco do Brasil com o seu aplicativo de chatbot que auxilia nos investimentos, avaliando algumas variáveis de entrada como valores da bolsa de valores, saldo em conta, entre outros.

Outro exemplo é de um trabalho realizado no Departamento de Física da Universidade PUC Rio apresenta as redes neurais temporais regulares e irregulares, conforme link http://www.puc-rio.br/pibic/relatorio_resumo2009/relatorio/fis/joao_pedro.pdf e imagens abaixo.



Métricas de avaliação

As métricas serão utilizadas pela porcentagem de cada captura de treinador, taxa de captura, evolução das características dos pokémons.

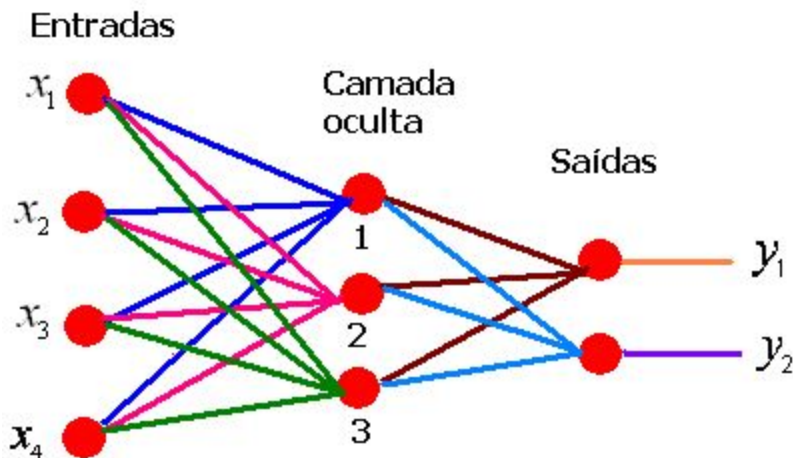
$$f(x) = (x - \min) / (\max - \min)$$

Neste caso, é a porcentagem de captura subtraindo pelo mínimo de captura de um pokémon, dividido pelo máximo de captura de um pokémon subtraído pelo mínimo de captura do treinador. Dessa forma terá a saída de ≤ 50 não capturou e > 50 capturou o pokémon.

Design do projeto

Serão utilizados os pontos abaixo:

- Cross Validation (Limpeza da base de dados)
 - Este processo será utilizado para avaliar e limpar as variáveis desnecessárias para o modelo e talvez seja utilizado o PCA para evitar muitas redundâncias no DataSet.
- Processo de batalha
 - Será utilizado o algoritmo de log loss, onde identifica a porcentagem de perda, o mínimo que deverá ser aceito é 0.06.
- Modelo de captura de pokémon
 - Roc curve será usado para avaliar o nível de interação entre as porcentagens, o que estiver acima de 0.5 será avaliado como assertivo e o treinador terá capturado o pokémon.
- PCA para não repetição do pokémon
 - O processo de não repetição do pokémon após um treinador já ter capturado um pokémon com o mesmo tipo e nome. Ele irá rejeitar a função de captura, onde será referenciado com (0 = não capturado e 1 = capturado).
- Processo de evolução do pokémon
 - Será utilizado o Backpropagation para o processo de evolução, já que o mesmo não é um processo linear, e será incluído o processo de punição para cada perda de pokémon ou batalha.



- Identificação do ganhador (Processo de predição)
 - Será utilizado o processo de rede neural temporal irregular, pois as variáveis são altamente modificadas em cada etapa do problema e para cada iteração deve ser levado em conta novamente o processo de predição.
- Liga pokémon
 - Será usado o Log Loss para validação de perda com a menor taxa entre cada treinador.

- Redes analógicas: processam dados de entrada com valores contínuos e, geralmente limitados. Exemplo deste tipo de rede são: Hopfield, Kohonen e as redes de aprendizagem competitivo.